CHALLENGES OF AGILE PRACTICES IMPLEMENTATION IN THE MEDICAL DEVICE SOFTWARE DEVELOPMENT METHODOLOGIES

Miloš D. Kostić
Faculty of Project and Innovation Management, Educons University, Serbia

Abstract: Agile practices have transformed many segments of the software development industry by streamlining the development process. Medical Device industry is facing new challenges as software is taking a more dominant role and the established development methodologies are struggling to meet the growing demand for rapidly produced high quality software. This paper presents review of the available literature on the subject of introducing the Agile practices in the Medical device industry with the aim to systematically point out the advantages, challenges and possible approaches to this process.Most challenges for embracing the Agile methodologies in the Medical Device industry seem to be rooted in the fact that the industry is regulated and Agile was not originally intended to address such processes. However, there are success stories from this and other regulated industries showing that implementation of selected Agile practices into the existing frameworks can be very beneficial.

Key words: medical device software, software development methodologies, Agile, safety-critical software

1. INTRODUCTION

Today, software is omnipresent in the modern world and it is almost impossible to imagine any segment without it. We use it to communicate, to shop, to organize our time and finances, to offer and contract different services, to entertain ourselves. However, it is also a part of all our electronic appliances, industrial plants, personal vehicles, public transportation, nuclear energy facilities, and life-sustaining Medical Devices. This trend generates a huge and growing need for development of new software. Furthermore, the exponential increase of market demand change rate is making technologies obsolete in an accelerated rate, pushing the companies to shorten their development cycles in order to keep up with the customers' requirements.

The sum of these factors created a need for methodologies that would allow faster and cheaper creation of application-specific software, which gave birth to Agile methodologies. For more than fifteen years many industries have been reaping the benefits of Agile methodologies (Jovanovic et al, 2016; VersionOne, 2010), however Medical Device industry is not one of them. Due to specifics of this industry there are many barriers to entry of new methodologies into this field.

The non-regulated software is developed with the aim to satisfy the customer's requirements, but safety critical software, such as Medical Device software, also needs to be developed in accordance with national and international regulations. The regulatory constraints are specific to the market for which the Medical Device software is planned. Regardless of the software being a part of a device, or a standalone product, it needs to comply with the local quality regulations, guidance documents and approved standards (Ge et al, 2010).

Medical devices in EU are regulated through the Regulation 2017/745 (European Commission, 2017) and the Medical Device Directive (MDD) 93/42/EEC (European Commission, 1993), which is amended by the Active Implantable Medical Device Directive (AIMDD) 90/385/EEC (European Commission, 1990), the In-vitro Diagnostic (IVD) Medical Device Directive 98/79/EC (European Commission, 1998), and updated with MDD 2007/47/EC (European

Corresponding author. Email: thekostic@gmail.com

ISSN2560-4961(online)

www.epmj.org

Commission, 2007). Similarly, in the US, this field is regulated by the FDA through the Code of Federal Regulation (CFR) Title 21, Chapter I, Subchapter H, Part 820 (FDA, 2017). These two systems, although different, are mostly aligned, which makes it possible for the same products to enter both markets. The devices are classified from Class I to Class III based on the specified use and the level of impactof the device on supporting or sustaining human life.

The software that has a role in a diagnostic process or administration of therapy, whether as an integral part of a device, or by itself, falls under these regulations. There are numerous standards and guidance that support manufacturers developing Medical Device software. Leveraging thesefacilitates adhering to the regulations, and expedites the process of entering these regulated markets. Some of the most notable standards used in the industry are ISO 14971 (2007), ISO 13485 (2003)], IEC 62304 (2006), IEC 62366 (2007), IEC TR 80002-1 (2009), IEC TR 80002-3 (2014a), IEC 82304 (2014b).

Typically, software developed for use within safety critical domains is developed usinglifecycles that have an upfront design, such as Waterfall Model or V-Model software development lifecycles (Bulska and Gorski, 2011). Their sequential structure places high importance on the production of documentation and ensures high traceability, which is why they produce the deliverables required by the regulations.Never the less, it is important to note that no regulatory requirements and development standards (ISO or FDA)specifically demand the use of any particularlifecycle when developing Medical Device software. In fact, they state that Medical Device softwarecan be developed using a traditional, iterative and/or evolutionary approach.

Even though the regulations and standards in the Medical Device industries are well-established, a significant percentage of Medical Devices is being recalled due to errors attributed to software. In 1990s the percentage of devices recalled due to software errors was around 10%, whereas in 2011 it reached 24%, and this growing trend doesn't seem to be changing(FDA, 2012). The increasing use of software in Medical Device industry is surely one of the main causesfor this growth, but it is also becoming clear that by relying on traditional methodologies it is not possible to adequately address the growing need for safe, yet rapidly developed Medical Device software

For example, the V-Model is a defacto industry standard.It is straight forward and produces necessary deliverables such as the documentation that ensures traceability between requirements and all stages of the software development lifecycle. Then again, it performs very badly in the event of a change once development has begun, as its rigid structure introduces high overhead in revisiting previous development stages, driving up the cost and delaying release dates (Munassar and Govardhan, 2010).

It would seem that introduction of Agile practices to the Medical Device software development is the logical next step. This review paper aims to determine the challenges of implementation of Agile practices in Medical Device software development life cycle, as well as to provide the answer to the question what are the potentially beneficial ways of addressing these challenges.

2. THE AGILE PRACTICES

The primary objective of Agile approach is to produce a working code as early and as efficiently as possible. In order to achieve that, the process is divided into iterations. Instead of setting a final project goal, each iteration has as an objective to produce the working code to primarily respond to the customer needs, and secondarily to the needs of the project. The advocates of the Agile processes often insist on the working code as the single important deliverable.On the other hand, according to them, the analysis and design models, as well as software documentation do not have such an important role in the development process. This is the point which is most often criticized among the Agile approach opponents. They stress that the consequence of the lack of solid documentation and models, especially when they are a part of large, complex systems, is quite often a corporate memory loss.

The application of such methodologies requires constant modifications and adjustments of the processes on both technical and managerial levels. These include adaptation to changes that emerge during the software development, software requirements, as well as the setting in which the development takes place. Nonetheless, Agileprocesses have become increasingly popular in the recent years. There's a significant number of methodologies described in the literaturethat are allegedly Agile. To avoid any kind of misconception in 2001 seventeen Agile process methodologists held a meeting where they defined precisely "agility" as a term. The result of the meeting was a publication of the manifesto of the "Agile Alliance" (see https://www.agilealliance.org/agile101/the-Agile-manifesto).

The "Agile Alliance" manifesto defines the core values of Agile Software Developmentas "i**ndividuals and interactions** over processes and tools", "**working software** over comprehensive documentation", "**customer collaboration** over contract negotiation", "**responding to change** over following a plan". It is important to note, that this does not challenge the value of the items on the right, just states that the value of the items of the left is greater. Furthermore, the manifesto clarifies corporate principles and suggests objectives for Agile processes through the twelve fundamental principles:

1. *"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."*
2. *"Business people and developers must work together daily throughout the project."*
3. *"Welcome changing requirements, even late in development."*
4. *"Deliver working software frequently."*
5. *"Working software is the primary measure of progress."*
6. *"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."*
7. *"The best architectures, requirements, and designs emerge from self-organizing teams."*
8. *"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."*
9. *"Agile processes promote sustainable development."*
10. *"Continuous attention to technical excellence and good design enhances agility."*
11. *"Simplicity is essential."*
12. *"Project teams evaluate their effectiveness at regular intervals and adjust their behavior accordingly."*

The potential benefits of Agile, as well as, the fact that there are difficulties in its implementation in the Medical Device Industry have been recognized both by the industry leaders and experts. The Association for the Advancement of Medical Instrumentation published a Technical Information Report (TIR) under the name TIR 45:2012--Guidance on the use of Agile practices in the development of Medical Device software (AAMI, 2012). The team in charge of the TIR development involved FDA staff and industry experts. The report emphasizes the gradual change in the non-regulated software development towards methodologies and practices that are rather Agile. It also recognizes the proof of successful implementation of Agile methodology in software development companies and organizations. However, the TIR authors claim that the details on the correct use of Agile practices are not comprehensive enough, hence their objective was to offer clear guidelines on the most adequate Agile practices that are to be applied in the Medical Device software development industry. Furthermore, it recommends the best practices for complying with international standards, as well as FDA guides when using Agile methodologies in the Medical Devices software development. Being a high-level document, the TIR report does not describe a large number of Agile methodologies in the development of software which complies with IEC 62304, and therefore does not offer guidance for the standalone software development.

3. AGILE PRACTICES AND MEDICAL DEVICE SOFTWARE DEVELOPMENT

As mentioned at the beginning, there is evidence of significant benefits being gained from implementation of Agile practices within the non-regulated software development industry, such as reduced time to market, costs reduction, and increased customer satisfaction(VersionOne, 2010). Still, Medical Device software development typically relies on a plan driven sequential Software Development Lifecycle (SDLC), such as the V-Model, mostly due to its alignment with regulatory requirements (McCaffery et al, 2005). This was confirmed in a recently published survey performed by McHugh et al. (2017), which found that around 75% of the organizations are developing software in conformity to some plan driven sequential SDLC, where the V-Model is represented in two thirds of the cases, and one third is relying on other traditional development lifecycles such as the Waterfall model. Nevertheless, it is worth mentioning that a quarter of the organizations is leveraging Agile practices to some extent. The same study confirmed the earlier reports that Medical Device software organizations are experiencing difficulties when following plan driven sequential SDLC, particularly in the area of requirements management. According to Mc Hugh et al. (2013) Medical Device software development overhead is constantly increasing and is primarily generated by introductions of changes after the beginning of development, as they result in revisiting stages of development.

Reports of these problems do not come as a surprise, as Royce, the father of the Waterfall model, noted the inherent problems associated with following a sequential lifecycle (Royce, 1987). Specifically, introduction of changes in requirements is increasingly difficult as the project progresses, since the requirements are fixed early on, and can create cost and budget overruns (Munassar and Goverdhan, 2010). On the other hand, capturing all of the requirements and properly prioritizing them at the beginning of the project can be very difficult (Cadle and Yeates, 2008).

As Agile methodologies are especially fit to address the changing requirement they seem to be a natural solution for this problem. The Mc Hugh et al. (2012)) identified 59 Agile practices and performed an analysis, which showed that none of the identified practices contradict regulations or development standards. However, only 13 practices have been successfully adopted in Medical Device software organizations developing regulatory compliant software.

Among the early adopters of Agile in the Medical Device industry there are success stories, which bring hope to the cause. Such is the case of Abbott Diagnostics published by Rasmussen et al. (2009). After the implementation of Agile practices within the plan driven SDLC on a development project in this company yielded between 35% and 50% of savings in cost, compared to the pure plan driven sequential SDLC.

Nevertheless, there are many practical issues in embracing Agile methodologies, which deter its widespread adoption. Some of these are discussed in the following segment.

4. THE CHALLENGES IN IMPLEMENTATION

As can be noted from reading the values and principles of Agile Manifesto, not all of it seems compatible with developing safety critical software, especially when compliance to regulations is based on traceability.

1. Regulatory limitation to tterative deployment

Iterative approach in which working versions of software with incremental improvements are delivered frequently is the backbone of the Agile approach. Thatway the customer can provide the feedback based on the firsthand experience with the actual product in the actual environment. This streamlines the development process and leads to much higher customer satisfaction but it is not compatible with safety-critical software regulations. According to both ISO 14971 (2007) and IEC 62304 (2006) the Medical Device(in this case software) must be fully tested and verified before it can be applied with patients. To perform this process within

every development iteration (which should vary from a few weeks to not more than few months) can actually increase the overhead and development costs.

2. Risk management and quality control

Another very important part of Medical Device software development is the risk management. A failure in this type of software can place patients, clinicians or other persons in a hazardous situation, which is why proper risk management procedures that ensure safety and reliability of the product are of paramount importance (ISO 14971, 2007).As noted by Boehm and Turner (2005), the Agile processes lack the structure and clear recommendations on how to systematically implement risk management plan in an efficient way, which can pose a barrier to adopting them in this industry. They further propose that the quality of the software produced following Agile methodologies is often lower than that produced in sequential plan driven models, which may additionally impede its adoption. Even though this statement may betrue for many use cases, it reflects more specific needs and focus of the industries operating in the "Internet time" than of the shortcomings of the used methodologies.

It is the fact though, that current Agile methodologies do not account for quality control mechanisms that are reliable enough and eliminate risk of direct human injuries and/or financial damage. Actually, there is a raising uncertainty that the existing quality control techniques such as informal reviews, pair programming and similar, would be sufficient in any safety critical domain, if applied alone. Use of techniques such as *Formal specification, rigorous test coverage, and other formal analysis and evaluation techniques* in the software development processdo provide a better guarantee for the safer use of the product, but also imply a significant mark up in cost (Turk et al, 2002).

3. Self-organized teams

To take full advantage of Agile practices it is important to allow the development teams to be self-organized, as stated in the eleventh fundamental principle. This however may pose a challenge for many organizations. It implies removing some of the decision-making powers from the management and can lead to management control loss that contradicts company policies (Moe et al, 2008). Furthermore, catering the specific needs of an Agile team can require significant changes in the Human Resource policies and processes (Boehm and Turner, 2005).

4. Documentation and traceability

One of the biggest strengths of Agile methodologies is the capability to accept and address requirements change even late in the development, making the requirements management much more efficient in comparison to the plan driven sequential SDLC models. Yet, this may pose the largest barrier to adoption of Agile practices. Regulations (FDA, 2002) stipulatethat the manufacturers provide thoroughly documented requirements before starting the implementation and the testing, as well as to have traceability between requirements and all stages of development. This can be considered as incompatible with the Agile'sfluid approach to defining the requirements progressively, "accepting and welcoming their changes even late in development". It also contradicts one of the core values, "*working software over comprehensive documentation*".

5. Distributed development environments

Nowadays, many industries operate on the global market and opt for global distribution of software development environments. This contradicts the postulate of the Agile methodology which states that the process is better performed if it is centralized in a single location. This way it allows for quicker and direct communication between team members and customers. Modern technology such as video conference calls can partially compensate for the lack of physical presence, however these technologies are almost always very expensive and their effectiveness is often overestimated. In both distributed and non-distributed environments, face-to-face communication is extremely valued, however in distributed environments it requires upfront planning, logistics and ensuring that all the

participants are available, and therefore it occurs less often. One should take advantage of these meetings to synchronize all the team members and participants in the project. Developers that are based in different geographical locations would have a chance to present their work and hear about the progress made by others, as well as discuss further steps and actions. In between these meetings the developers can communicate through the software documentation. Clear requirements and designs presented in this documentation and their delivery in a timely manner is an imperative and the only way for geographically distributed project participants to have the identical product vision. However, this doesn't mean that every single step in the project development must be strictly documented. Documented and modeled should be only those aspects of software that bring value to the project and its stakeholders (Turk et al, 2002).

6. Limited support for building reusable artifacts

The Agile doctrine (for example: Extreme Programming process) recommends creating software products that would solve a concrete problem. Despite the fact that building generalized solutions sometimes has long-term benefits, developing software in "Internet time" often doesn't allow for that luxury. In these instances, the best way to develop such generalized solutions and reusable software forms (for example: design frameworks) is through projects whose main focus is building and development of reusable artifacts. Researchers at the University of Maryland at College Parkhave created a framework entitled "The Experience Factory" (Basili and Rombach, 1991), which is reuse-oriented and focuses primarily on the split between the development of product-specific software and building the reusable artifacts. For the reusable artifacts to be widely applied the strict quality control is an absolute imperative, since it is estimated that the impact of the poor quality such as severe errors is equal in percentage as the number of times that artifact is being reused. It is also desirable that these reusable artifacts are delivered in a timely manner. However, it remains unsolved how Agile methodology can be adequately adapted in the development of reusable artifacts.

5. EXPERIENCES FROM OTHER REGULATED INDUSTRIES

As most barriers to implementation of Agile practices seem to be related to the regulated nature of the Medical Device Industry lessons learned from within other regulated safety critical industries, such as Avionics and Automotive, can shed some light on potential solutions and pitfalls on this path.

Wils et al (2006) investigated the possibility of implementation of extreme programing in the avionics industry. The authors argue that avionics software development could benefit from adopting Agile practices, despite the fact that in the case study conducted such approach turned out not to be practical. They suggest thatthe largest potential gain could be achieved in the requirements managements and change management, however, only if some prerequisites are met, e.g. early customer involvement. This claim is supported by the results from Vanderleest and Butler (2009) who investigated the compatibility between Agile practices and the development practices proposed by the relevant standards in the Avionics industry. According to their research implementation of these practices would be possible and beneficial, however at that time the subject was still insufficiently explored in the Avionics industry research community.

The automotive industry shares similar regulatory constraints and faces similar challenges as software is becoming ever more present in all of its segments. In a case study presented by Manhart and Schneider (2004) the possibility of transferring the operations of engineering department of Daimler Chrysler to full Agile methodology was investigated. Their results indicate many of impedimentsfor "going full Agile" listed in the previous section making, but they found that a hybrid framework tailored to their specific requirements gave positive results. They managed to integrate some Agile practices, like test first process, into their process improvement methodology, that relied on a traditional model.

6. PROPOSED APPROACHES

Current regulations for Medical Device software development and the accompanying standards do not stipulate use of any specific software development methodology, nor do they forbid any. There are specified deliverables that the manufacturer must produce and provide in order to be permitted to enter the regulated market, but the choice of development lifecycle used to create those is not limited. Therefore, the question of embracing Agile practices is more a practical than a regulatory one. According to Rottier et al. (2008) none of the currently used Agile methodologies can be strictly implemented when developing Medical Device software, as no single Agile methodology provides the framework to address all of the segments needed to comply with the regulations. However, they did find that integrating selected Agile practices into plan driven SDLC can greatly increase the efficacy of the development process, while keeping the conformity with all the relevant regulations and standards. These findings were also confirmed by Rasmussen et al. (2009).

Turk et al (2002) identified some of the Agile practices that can be considered favorable to the creation of Medical Device software: *(1) test-first approaches requires one to define unit tests before writing code, (2) the early production of working code supported by the incremental, iterative process structure of Agile processes supports exploratory development of critical software in which requirements are not well-defined, and (3) pair programming can be an effective supplement to formal reviews.* They suggest that the Agile software development can actually complement the formal software development if needed. Formal approach can be performed in an Agile manner when handling critical pieces of the software, to achieve higher performance and efficiency, while maintaining the required quality and confidence.

The similar conclusion was reached by Mc Hugh et al (2013) who proposed an Agile hybrid model based onthe V-model with the following rational:

- *Medical device software organizations typically follow the V-Model to develop Medical Device software. As a result, they are already familiar with the structure and phases of the V-Model*
- *Medical device software organizations may have already received regulatory approval to follow the V-Model when developing Medical Device software.*
- *Whilst none of the regulatory requirements or development standards mandate the use of the V-Model, it appears to be the best fit with regulatory requirements, as it guides organizations through the process of producing the necessary deliverables required to achieve regulatory conformance.*

They also propose adding iterative Risk identification to the model, a process in which the project is divided in segments and for each segment a risk assessment is performed. Their recommendation is to prioritize the execution of those segments what pose the most risk.

7. CONCLUSION

The Medical Device software development is specific,as it needs tosatisfy both client needs and regulatory requirements. This literature review has addressed the obstacles of employing Agile practices to streamline this process, making it more suitable to rise up to the modern challenges of accelerated technology lifecycle. Even though Agile seems to be an ideal solution for most applications, many of its virtues turn to impediments when applied to the Medical Device industry. Regulatory requirements are not compatible with full implementation of Agile principles of iterative development with frequent releases of functional software, nor with fluent changes of requirements, which hinders traceability. Furthermore, current Agile frameworks lack the support for appropriate quality control and risk management. On the other hand, there are organizational issues to embracing some Agile practices, such asintroduction of self-managed teams, as well aslimitedsupport of Agile methodologies for working with

geographically distributed development teams, or production of reusable artefacts. This is why adoption of handpicked Agile practices that complement existing SDLC may prove to be most beneficial approach. In this way, a well- defined SDLC for which the development companieshave already received regulatory approval do not need do suffer major changes, especially in the safety critical aspects.

REFERENCES

Wils, S. Baelen, T. Holvoet, and K. Vlaminck, (2006) "Agility in the avionics software world," presented at the 7th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP.

AAMI, A. T. (2012). Guidance on the use of Agile practices in the development of Medical Device software. Association for the Advancement of Medical Instrumentation, Arlington, VA.

Basili, V., & Rombach, D. (1991). Support for comprehensive reuse. Department of Computer Science, University of Maryland at College Park. UMIACS-TR-91-23, CSTR-2606.

Boehm, B., & Turner, R. (2005). Management challenges to implementing Agile processes in traditional development organizations. *IEEE software*, 22(5), 30-39.

Bulska, K., & Gorski, J. (2011). Applying Agile practices to the development of safety-critical software. ICT Young 2011, Scientific Booklets of Faculty of Electronics, Telecommunications and Informatics, 65-68.

Cadle J. and Yeates D. (2008), Project Management for Information Systems: Pearson Education.

European Commission, Council directive 90/385/EEC on active implantable Medical Devices (AIMDD). 1990: Brussels, Belgium.

European Commission, Directive 2007/47/EC of the European Parliament and of the Council concerning Medical Devices, OJ no L247 2007-09-21. 2007, EC: Brussels, Belgium.

European Commission, Directive 93/42/EEC of the European Parliament and of the Council concerning Medical Devices, in OJ o L 247 of 2007-09-21. 1993: EC, Brussels, Belgium.

European Commission, Directive 98/79/EC of the European parliament and of the council of 27 October 1998 on in vitro diagnostic Medical Devices. 1998: Brussels, Belgium.

European Commission, Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on Medical Devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC Official Journal of the European Union L 117/1 5-5-2017. Web site. http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex:31993L0042 Accessed December 10, 2017

Food and Drug Administration, 2017. Chapter I - Food and drug administration, department of health and human services subchapter H - Medical devices, Part 820 - Quality system regulation. [cited April 2017]; Available from: http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820.

Food and Drug Administration. FDA News on Software Failures Responsible for 24% of all Medical Device Recalls. 2012 [cited 2015 06.03]; Available from: http://www.fdanews.com/newsletter/article?articleId=147391&issueId=15890.

Food and Drug Administration. (2002). General principles of software validation; final guidance for industry and FDA staff. Food and Drug Administration, 11.

Ge, X., Paige, R. F., & McDermid, J. A. (2010). An iterative approach for development of safety-critical software and safety arguments. In AGILE Conference, 2010 (pp. 35-43). IEEE.

IEC 62304:2006, Medical device software—Software life cycle processes. IEC: Geneva, Switzerland. (2006)

IEC 62366:2007, Medical devices - Application of usability engineering to

Medical Devices. IEC: Geneva, Switzerland.

IEC/CD 82304:2014, Health Software - Part 1: General Requirements for Product Safety. ISO: Geneva, Switzerland. (2014)

IEC/TR 80002-1:2009, Medical device software Part 1: Guidance on the application of ISO 14971 to Medical Device software. BSI: London. (2009)

IEC/TR 80002-3:2014, Medical Device Software - Part 3: Process reference model for Medical Device software life cycle processes (IEC 62304). ISO: Geneva, Switzerland (2014)

ISO 13485:2003, Medical devices — Quality management systems — Requirements for regulatory purposes. ISO: Geneva, Switzerland. (2003)

ISO 14971 (2007) Medical Devices — Application of risk management to Medical Devices. ISO: Geneva, Switzerland.

Jovanović, A. D., Jovanović, F. P., Miletić, L. Z., &Berić, I. M. (2016). Application of agile methodologies in software development. *Tehnika*, 71(6), 896-900.

Manhart, P., & Schneider, K. (2004). Breaking the ice for Agile development of embedded software: An industry experience report. In Proceedings of the 26th international Conference on Software Engineering (pp. 378-386). IEEE Computer Society.

Mc Hugh M. (2012) "Integrating Agile Practices with Plan-Driven Medical Device Software Development," presented at the The 13th International Conference on Agile Software Development: XP 2012 Doctoral Symposium, Malmo Sweden.

Mc Hugh, M., Cawley, O., McCaffcry, F., Richardson, I., & Wang, X. (2013). An Agile v-model for Medical Device software development to overcome the challenges with plan-driven software development lifecycles. In Software Engineering in Health Care (SEHC), 2013 5th International Workshop on (pp. 12-19). IEEE.

Mc Hugh, M., McCaffery, F., & Casey, V. (2017). Barriers to using Agile software development practices within the Medical Device industry.

McCaffery, F., McFall, D., Donnelly, P., Wilkie, F. G., &Sterritt, R. (2005, April). A software process improvement lifecycle framework for the Medical Device industry. In Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the (pp. 273-280). IEEE.

Moe, N. B., Dingsøyr, T., & Dybå, T. (2008, March). Understanding self-organizing teams in Agile software development. In Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on (pp. 76-85). IEEE.

Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *IJCSI International Journal of Computer ScienceIssues*, 7(5), 94-101.

Rasmussen, R., Hughes, T., Jenks, J. R., & Skach, J. (2009). Adopting Agile in an FDA regulated environment. In Agile Conference, 2009. AGILE'09. (pp. 151-155). IEEE.

Rottier, P. A., & Rodrigues, V. (2008). Agile development in a Medical Device company. In Agile, 2008. AGILE'08. Conference (pp. 218-223). IEEE.

Royce, W. W. (1987, March). Managing the development of large software systems: concepts and techniques. *In Proceedings of the 9th international conference on Software Engineering* (pp. 328-338). IEEE Computer Society Press.

Vander Leest, S. H., & Buter, A. (2009). Escape the waterfall: Agile for aerospace. In Digital Avionics Systems Conference, 2009. DASC'09. IEEE/AIAA 28th (pp. 6-D). IEEE.

VersionOne, (2010) "State of Agile Survey - The Stage of Agile Development,"